# EXAMINING THE COMPLEXITY OF POPULAR WEBSITES

by

RAN TIAN

A THESIS

Presented to the Department of Computer and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

June 2015

THESIS APPROVAL PAGE

Student: Ran Tian

Title: Examining the Complexity of Popular Websites

This thesis has been accepted and approved in partial fulfillment of the requirements for the Master of Science degree in the Department of Computer and Information Science by:

Reza Rejaie Advisor

and

Scott L. Pratt Dean of the Graduate School

Original approval signatures are on file with the University of Oregon Graduate School.

Degree awarded June 2015

THESIS ABSTRACT

Ran Tian

Master of Science

Department of Computer and Information Science

June 2015

Title: Examining the Complexity of Popular Websites

A significant fraction of today's Internet traffic is associated with popular web sites such as YouTube, Netflix or Facebook. In recent years, major Internet websites have become more complex as they incorporate a larger number and more diverse types of objects (e.g. video, audio, code) along with more elaborate ways from multiple servers. These not only affect the loading time of pages but also determine the pattern of resulting traffic on the Internet.

In this thesis, we characterize the complexity of major Internet websites through large-scale measurement and analysis. We identify thousands of the most popular Internet websites from multiple locations and characterize their complexities. We examine the effect of the relative popularity ranking and business type of the complexity of websites. Finally we compare and contrast our results with a similar study conducted 4 years earlier and report on the observed changes in different aspects.

CURRICULUM VITAE

NAME OF AUTHOR: Ran Tian

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene

DEGREES AWARDED:

Master of Science, Computer and Information Science, 2015, University of Oregon
Bachelor of Science, Computer and Information Science, 2012, University of Oregon

AREAS OF SPECIAL INTEREST:

Networking Measurement

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to Professor Reza Rejaie for his long-term assistance in this research. I also sincerely thank to, those who collect data from different locations make significant contribution to our project. Juan Miguel Carrascosa Amigo from Spain, Balarishman Chandrasekaran from Duke, Roberto Gonzalez from France, Flavio Vinicius from Brazil and Xu Yuan from China provide valuable data for us to proceed further analysis. Reza Farahbakhsh from France also helps developing detail process in collecting data. They are highly cooperative and friendly to our project. I also thank for the help from Bahador Yeganeh and Motamedi Reza from University of Oregon. They provide valuable resources that help in improving the results.

TABLE OF CONTENTS

Chapter Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

# CHAPTER I
# INTRODUCTION

Nowadays, many famous websites have successfully become indispensable in human's life, such as Facebook and Amazon. Although more non-web based services step in, e.g. iOS apps, using the web browser is still the most convenient and efficient way to link to the online society. Web pages carry all objects that contain information. With the development of web techniques, more and more elements are added to web pages for all kinds of purposes. We observe that such colorful web pages may bring technical issues from the view of server side. A web page usually requires distributed servers to render different objects. In addition, the sophisticated networking communications may cause connection or speed issue between the client and servers.

Well-known websites always have advantages on loading web pages. They have more servers, larger bandwidth and well-organized web pages. The first two usually cannot be controlled by small entities, such as persons or small companies, since they are limited in resources, funding and locations. The web page, however, usually determines the probability of attracting people. On the other hand, websites focus on different objects based on their types. News websites provide more text info; business websites usually have more content on CSS styles; and sports websites need to render a number of animations. Different orientations make websites distinct. Users understand the differences. Hence, they are more tolerant toward slower loading speed of espn.com than google.com.

A valuable report [1] reveals the correlation between websites complexity and performance of their home page. The authors studied the combination of complexity and performance from websites to present result-driven conclusions. They collected data of approximately 1700 websites from multiple vantage points around the world. It was found that the correlation between performance and objects, servers and Multipurpose Internet Mail Extensions (MIME) types affect page loading time. This may help web developers better understanding how the users experience their services. Small entities may also adjust their websites to maximize the benefits.

Over years, the expansion of the network may have changed the situation. More websites, more contents and more bandwidth may take slightly different impact on users' experiences. The page loading time may actual extend slowly, though people do not

obviously notice. Our goal is to re-examine the environment of Internet society and revise the previous methodology so that people can get up-to-date information for more benefits. We select several vantage points from major Internet usage places, and use a major point as our focus. Each vantage point works individually simulating normal visiting towards famous websites with records. We collect the record data and organize them with appropriate format to analyze most points focused by the previous work and extend them with more details. Therefore, we can show the correlation between complexity and performance according to nowadays Internet environment; and compare and contrast with previous work to understand the variation of the Internet.

As results, our study find that websites are loading more objects, spending more time and contacting more servers. The improvements vary around 50%. More popular websites render more content, although with longer page loading time, according to [1] and our results. However, the origins vs. non-origin servers are quite different. In terms of numbers are quite unbalanced. The number of origins shrinks significantly, while non-origin servers greatly increase. According to empirical discovery and evidence, we believe the method distinguishing origins and non-origin servers is not as reliable as years before. On the other hand, using number of requests or servers to predict total page loading time is still reasonable. The more objects or servers requested, the more time that a web page needs to finish loading.

In this paper, we begin with introducing our measurement setup in Chapter II. We present the selection of websites, vantage point and crawling process in details. Chapter III gives a general overview of our data set. We talk the statistics of our final data and how we handle the errors. Then, we move into the details of our results. We picture the major complexity analysis in Chapter IV. Chapter V focuses on the performance issue connecting with our complexity studies. Chapter VI summarizes our results and concludes the strong and weak points of our research.

# CHAPTER II
# METHODOLOGY

## Websites

There are several famous websites providing ranking service for all websites around the world on the Internet. We select alexa.com and quantcast.com as two main references. The alexa.com run by amazon.com, provides websites ranking services and categorizes each website with reasonable labels. The quantacast.com ranks approximately a million websites. We use these two ranking lists to analyze and validate the pattern of complexity and performance results.

However, since these two ranking lists have great differences, we decided to use both lists. The quantacast.com ranks nearly 1 million websites. Examining all of them is far beyond the scope of our study. In fact, we believe that sampling the top 20,000 websites from quantacast.com is sufficient since these websites draw most attentions on the Internet. Specialized websites are usually less known and thus have a smaller group of users. As a result, from the quantacast.com, we select different number of websites from different ranking range. Table 1 shows the plan of result. We select all top 500 websites from quantacast.com. We also randomly select 300 from 500 - 1,000 and 300 from 1,000 - 5,000. Later, we pick up 400 from 5,000 - 10,000 and 500 websites from 10,000 - 20,000. Although, it seems we select more websites from lower rank, the distribution of websites across the top 20,000 ranking is still high rank biased.

| Rank Range | Quantacast Websites | Alexa Websites |
|---|---|---|
| 1-500 | 500 | 344(134) |
| 500-1000 | 300 | 11 |
| 1,000-5,000 | 300 | 54 |
| 5,000-10,000 | 400 | 37 |
| 10,000-20,000 | 500 | 44 |
| Total | 2000 | 500 |

**Table 1. Details of Websites Distribution across Ranking Ranges**

On the other hand, we also select 500 websites from alexa.com. However, we are

intentional to use these websites directly, since they are ranked differently in quantacast.com. So, we check and measure these websites according to quantacast.com. Table 1 shows the number of websites shown in each ranking range of alexa.com according to quantacast.com. However, there are still 134 websites from alexa.com have no rank in top 20,000 in quantacast.com. We select them as priority to rest of websites. In other words, we take away around 31 - 32 websites from quantacast.com in each range from 500 - 20,000. We call the result of the final selection as master list.

## Process

As [1] suggested, we also use a browser to simulate human action. We notice that Selenium WebDriver [2] provides good support for automatic browsing, so we use the library to operate JAVA based scripts. The scripts use the build-in version of Firefox [3] provided by Selenium WebDriver. Then we use the extension of Firebug (version 2.0.2) [4] with Firestarter (version 0.1a6) [5] and NetExport (version 0.9b6) [5] to automatically output the HTTP archive record (HAR) files [6] for detail analysis after page loading is completed. The HAR files follows JSON format as a popular NoSQL data format. They record all the details of requests and responses that the browser sends after the structure of target page is loaded, including all kinds of timestamps, content of requests and responses and etc. They provide valuable information for parsing the web pages.

Another open source library called HarLib [7] based on Jackson parser is also used in our research. It extracts the documents of the HAR files into Java objects in order to conduct further analysis.

We organize our analysis into two parts: crawling and parsing. The first phase of crawling visits all pages that are chosen and generates HAR files corresponding to each websites. The second phase of parsing, after finishing the first phase, reads all the generated HAR files, parses and assembles them into our databases for our final analysis.

### Crawling:

The crawling initiates with a parameter file, which stores all the websites from master list that may potentially be visited. Each vantage point visits the same master list with different ordering. There are no outliers for vantage points. In other words, each vantage point visits 1000 or 2000 fixed URLs only once.

The Internet may redirect our visit. Since some websites own multiple domain names,

for example, yahoo.com is redirected to www.yahoo.com, and bilibili.tv is redirected to www.bilibili.com, we consider such behaviors as normal and ignore the differences between the initial and final URLs. However, some websites own different websites for unique initial URL and ranking list consider the differences as different websites. An obvious example is google.com. If we visit google.com from Japan, which ranks first in Quantacast.com, for instance, we are redirected to google.co.jp, which has its own ranking of 34732 in Quantacast.com as well. Therefore, we cannot consider such redirection as normal since we actually visit different websites (google.co.jp considers Japanese as its priority language, while google.com uses U.S. English). We categorize them by their final URLs.

Cycles: we define a complete visit of a specific web page as a cycle. The Selenium WebDriver runs a cycle by initiating a new instance of Firefox process with our specified preferences. Then the Firefox load the web page by its full domain name according to alexa.com. The page may redirect to other domain name under certain situation, e.g. http may be redirect to https and google.com may be changed to www.google.com/?gws_rd=ssl. The page itself must be primitive as well, which means it does not contain any cookies, credentials or confidential information, e.g. facebook.com asks a login or registration instead of displaying someone's home page. After certain amount of time, the Firebug output the HAR file and the instance of Firefox is completely shut down to avoid any cookies or cached content for subsequent cycles.

**Parsing:**

The parsing is more straightforward than crawling. The parser takes each HAR file that Firefox generated reads and stores the data into database according to the vantage point and ranking list by linking them with their rankings and final URLs. As we introduced before, we are satisfied to parse the HAR file by HarLib. The time that each parsing takes varies due to its content length, but the total time for parsing 1000 - 2000 websites is about hours.

## Vantage Points

In this paper, we aim to simulate that a "common" user visits popular website from our vantage points. The selection of vantage points is considered based on our two major purposes: complexity and performance.

First, ideally, the content of a specific website should be similar even if we visit from

different vantage point at different time. The fact is that the pages actually change over time. Websites often manage updates in an expedited fashion. For example, yahoo.com has to update its news regularly in order to function as an up-to-date news website. On the other hand, the geographical position of vantage point also matters. The pages that we reach from different vantage points can be slightly different, e.g. local news section is changed. However, we believe that a certain website should provide similar information in terms of complexity during a reasonable short period of time, if the actual page we reach is the more or less the same one, e.g. no redirection to other language or region based page with totally different URLs. Therefore, we believe a vantage point is enough to gather valuable information of web pages.

The performance, however, changes greatly if the vantage points highly diverse. The round-trip time (RTT), reachability and content itself could easily result in different outputs. We should select different vantage points in terms of fairness. The results should be statistically consolidated to get an unbiased conclusion.

We use our own location as main vantage point during the process. We also perform several runs when collecting. According to [1], since different vantage points provide about the same results, we believe the result from one single vantage point is also valid to present most information.

In addition, we select several vantage points from major Internet usage area as shown in Table 2. LocationID indicates the country code and date that the crawling process takes. For example, US-W-5-02 names the location in west of U.S. and the data on May 2. We deploy our crawler on each vantage point. Each vantage point takes a small run of 1000 websites in total. Each crawler runs independently, and generates corresponding HAR files.

## Parameters

After a general discussion about our process, we now go into details. Each part of our process has one or more undermined parameters, which may change the results with different settings.

**Complete Page Loading:**

The first question we encountered is how to determine page loading is completed. A common sequence of page loading is to send a request for the base page first. Then for each object that base page contains, sending request with certain strategy. The principle is to send

requests as many as possible to minimize page loading time. In practice, however, it is difficult for the browser to do so. If requests have dependencies or the maximum number of simultaneous requests is reached, the request may be delayed until previous work is complete. Such pattern presents a cluster of requests and responses in time-line. Some web pages request a constant update request initiated by the client, which means after a certain amount of time, the browser sends a request to update page content. However, the Firebug requires a timeout to determine the completion of page loading. In other words, if the browser does not send any request after the last response from the server, the Firebug concludes the completion of page loading and outputs the HAR file.

| LocationID | City/State | Country | Continent |
|---|---|---|---|
| US-W-5-02 | Eugene, OR | U.S.A. | N America |
| BRA-2-15 | UFMG | Brazil | S America |
| FRA-2-09 | Paris | France | Europe |
| US-E-2-09 | Durham, NC | U.S.A. | N America |
| SPA-2-12 | Madrid | Spain | Europe |
| CHN-2-15 | Shenzhen, Guangdong | China | Asia |

**Table 2. Details of Vantage Points**

Therefore, if the value of timeout is too large to avoid an update request, the Firebug never gets to the end of page loading. If the value is too small, the firebug may finalize the page loading period before the actual page is completely loaded. Unfortunately, there is no straightforward guidance for the timeout value, since the performance measurement could greatly influence the time-line of page loading; also a constant value is not feasible to reach due to dynamic environment of network bandwidth. We empirically use 10 seconds for the timeout value.

**Maximum Cycle Time:**

Cycle may be not completed due to a variety of reasons. The direct result of an incomplete cycle is no output of HAR file. The Firefox may also get stuck thus next cycle may fail to initiate. However, there is no reason to stop consecutive loading of cycles. Therefore, we define 60 seconds as the maximum cycle time. If a cycle exceeds maximum

cycle time, within a 15-second grace period, either the thread of Selenium WebDriver or the process of Firefox should be terminated. Both methods lead to a contiguous cycle that can finish our work eventually.

**DNS LOOKUP:**

In order to understand different servers, we use dig tool to query the Domain Name System (DNS) to obtain information of the server during our parsing process. We called such operation as *dnslookup*. However, dig uses unreliable connection (UDP) to obtain data as default. Therefore, *dnslookup* operation may result in invalid response, e.g. timeout due to packet loss. We allow 2-second retry and at most 2 attempts for each operation.

CHAPTER III

DATA SET

## Errors and Handling

We face multiple errors in both phases of our analysis. We introduce these errors and discuss their effects and our mechanism to process them.

**Unreachable Websites:**

Not all websites are available at any moment. It is surprising to find some websites, especially some of which are among the top 500 most popular list, are outage for weeks. Also out vantage points may have network connectivity issues. Target domain may timeout responses; some requests for certain objects may get no response. All these circumstances lead to incomplete page loading. In addition, Selenium WebDriver may report unexpected internal errors, which may fail to initiate a Firefox instance or execute reaching target domain. We ignore any form of unexpected error during web page visiting process as if we never set these websites as one of our target. They do not count towards any result or conclusion.

Such method may lead to a biased result since some websites that in common sense are considered very famous may be neglected as part of errors. However, we believe that errors less than 5% of all visiting are reasonable and acceptable. Our major results are still powerful and valid.

**Missing, Incompatible or Corrupted HAR Files**

There are many possibilities to find missing, incompatible or corrupted HAR files during the phase of parsing. The Firefox may generate an empty HAR file if the website is unreachable and Firefox stays in idle as if it finishes page loading, and the crawler still link the empty HAR file to that website. Not all responses from the server are compatible with the Firebug extension and Firebug may also generate incompatible HAR file with our HarLib parser.

Most HAR file errors are handled automatically. Blank HAR file contributes nothing to our analysis as if the crawler never visits the website. Corrupted or incompatible HAR files caught by HarLib would return in exceptions, which are handled internally. All exceptions are not rendered in our output as well. So HAR file errors are not taken into consider and exert no impact on any results.

## Result

In the crawling phase, a total run on one vantage point lasts 60 seconds for each website and 2000 websites take approximately 20 hours. Different vantage points result in different data. The detail of parsed data is shown in Table 3. Rounds column shows the number of rounds completed in corresponding location. Planned column shows total websites that each vantage point plans to visit. Collected column tells how many websites generate HAR files, which require complete crawling cycles. Conflict column shows how many websites visit the same final URL due to redirection. We exclude these websites in case of fairness. Unresolved column indicates the number of errors reported by our parser. Parser may throw exception if any error occurs and abandon parsing so that certain HAR files are not included into our final statistics. Real column indicates the number of HAR files that we finally take.

| LocationID | Rounds | Planned | Collected | Conflict | Unresolved | Real |
|---|---|---|---|---|---|---|
| US-W-5-02 | 4 | 2000 | 1793 | 23 | 58 | 1712 |
| BRA-2-15 | 1 | 1000 | 966 | 12 | 25 | 929 |
| FRA-2-09 | 1 | 1000 | 944 | 11 | 20 | 914 |
| US-E-2-09 | 1 | 1000 | 904 | 11 | 27 | 866 |
| SPA-2-12 | 1 | 1000 | 960 | 12 | 25 | 923 |
| CHN-2-15 | 1 | 1000 | 841 | 53 | 125 | 663 |

**Table 3. Details of Data Collected According to Vantage Points**

Most vantage points have around 90% valid reports. One interesting finding is the data from China. Due to censorship, many websites in top rank lists are unreachable, e.g. google.com. Most of them are blocked, which result in either bad access or connection reset. Sometimes, even valid connection may timeout due to bad network environment. Therefore, results from China are abandoned in our project because valid sample is too small in quantities.

# CHAPTER IV

# COMPLEXITY

## Overview

Similar to [1], we investigate most results by website ranking and category. Ranks are collected from quantacast.com. However, some websites are redirected to some other destinations that are not ranked in top 20,000 in quantacast.com. We are unable to associate rank with them. Therefore, they are not considered by ranking.

Each website may belong to one or more corresponding categories. There are 17 first level category labeled by alexa.com. We mark them as main categories. alexa.com also provides second level categories, which are subcategory of main categories; and the third level categories are also attached to second level; and so for. A single website can be labeled in details. For example, jenniferthieme.com can be found under category of Business > Accounting > Firms > Bookkeeping and Tax Preparation > North America > United States > California.

However, alexa.com does not share the information directly. Each category only shows up to 500 websites. In addition, subcategories are overlapped according to their parent categories. We only parse main and second level categories. If any websites are listed in more than one category, we use the more popular one, which has the larger number of websites contained.

Table 4 shows the number of websites label in corresponding ranking groups. Note that some websites from quantacast.com do not have a corresponding rank in top 20,000 quantacast.com websites. They are not listed in the table. Although there are 17 main categories, we do not consider the categories that contain less than 50 websites as [1]. Since not all websites from quantacast.com are categorized on alexa.com, the majority of data entries are N/A.

| Rank | Art | Business | Computer | News | Shopping | Society | Sports | N/A | Total |
|---|---|---|---|---|---|---|---|---|---|
| 1-500 | 17 | 11 | 19 | 6 | 13 | 11 | 14 | 57 | 170 |
| 500-1k | 8 | 11 | 4 | 4 | 11 | 13 | 7 | 78 | 143 |
| 1k-5k | 13 | 16 | 15 | 7 | 22 | 13 | 7 | 155 | 267 |
| 5k-10k | 12 | 21 | 9 | 7 | 16 | 8 | 6 | 242 | 339 |
| 10k-20k | 11 | 25 | 15 | 5 | 14 | 17 | 8 | 345 | 460 |
| Total | 61 | 84 | 62 | 29 | 76 | 62 | 42 | 877 | 1379 |

**Table 4. Category Distributed According To Ranks**

### Requests

We first take a look at number of requests sent to servers. This can be also considered as the number of objects on each front page of websites. A website usually loads a page frame at the beginning. Then, it renders different objects according to the frame. Figure 1 shows the cumulative distribution function (CDF) of number of requests across all websites by rank. We can assume a typical normal distribution, which is around 50 to 300 requests that each website may send. A big portion of approximately 80% websites send less than 200 to 250 requests by corresponding ranking. It also means a considerable number of websites that have abundant contents reaches more than 400. The ranking is also distinguishable. Websites with higher ranks normally load more objects than lower ranks. Figure 2 narrows down the requests by category. Most categories share similar curve, while sports and arts websites show obvious diversity. Interestingly, [1] also finds a different category of News. We believe the similar reason as [1], since many such websites, e.g. espn.go.com, load abundant contents in homepage; and they usually do not require private sessions such as login requirement.

An apparent difference between our figure and the figures provided by previous work [1] is the right shifting of curves. The number of requests in each range increases significantly. About 50% more requests are found in our result. It means more objects are requested visiting websites nowadays. We believe such growth is reasonable and inevitable because of increasing networking traffic load.

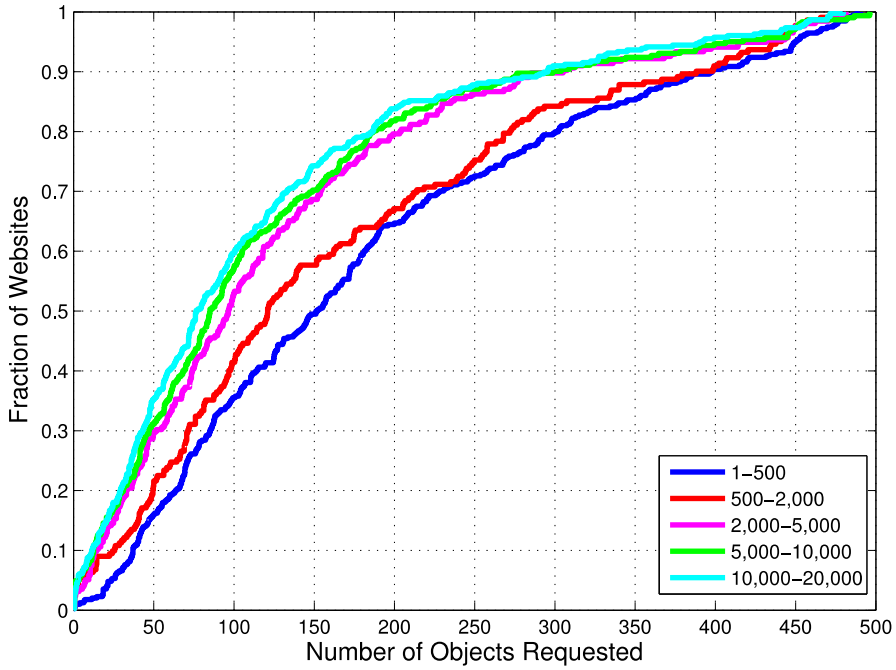


**Figure 1. Cumulative Distribution Function (CDF) plot of total number of objects requested across all websites by rank**

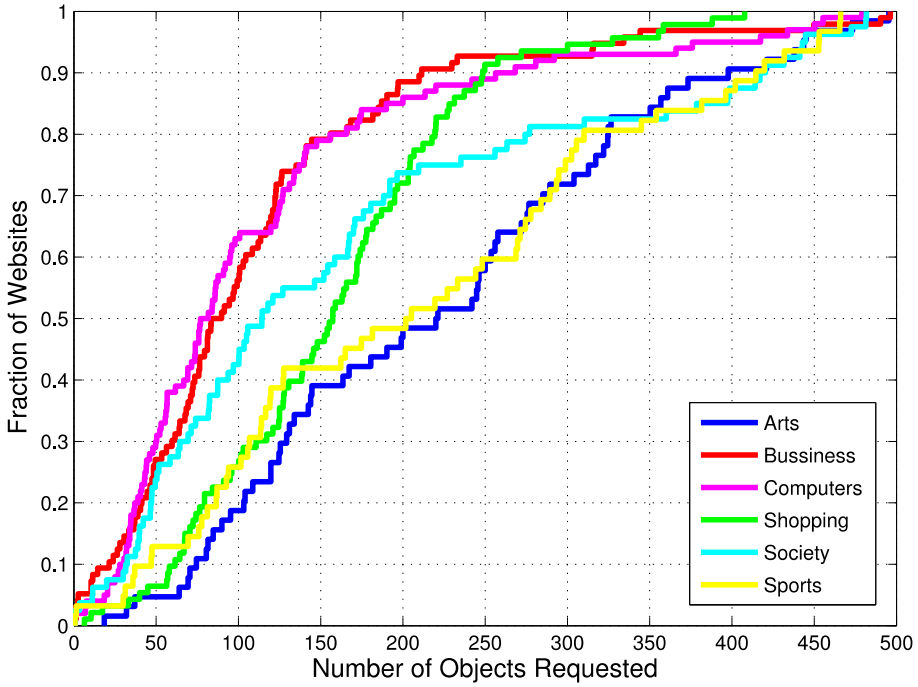


**Figure 2. Cumulative Distribution Function (CDF) plot of total number of objects requested across all websites by Category**

**MIME Types**

Number of objects: then we take a look at the loaded objects. Different types of objects are identical to each other. For instance, images are usually larger than CSS styles. We categorize each object by its MIME type. Table 5 shows the definition of different MIME types that we select to compare according to official registration by IANA. The column of composition indicates the proportion of corresponding type in page loading process in numbers and size. In other words, for example, images occupy 48.51% in number of requests and 49.39% in size of content.

| Type Name | Composition (Number/Size) % | MIME label in HTML | Notes |
|---|---|---|---|
| image | 48.51 / 49.39 | image/* | images |
| JavaScript | 18.88 / 17.43 | */javascript | JavaScript |
| CSS | 3.82 / 2.92 | */css | CSS style |
| Flash | 1.20 / 7.80 | */x − flv or */x − shockwave − flash | Flash application |
| html-xml | 15.82 / 2.91 | application/xml | Extensible Markup Language |
| text | 3.36 / 0.49 | text/* | any form of text info |
| other | 8.39 / 19.92 | N/A | any type that does not belong to previous category |

**Table 5. MIME Type Split and Notes**

Figure 3 shows the distribution of requests of different MIME types of content by rank. Notice that there are more than 100 different types including official MIME types, which are defined in RFC standards, and experimental or non-official types. The results confirm that a majority of image requests take up the large portion the page loading time. The median is around 50 and third quartile exceeds 100. Similar to total requests, we find decreasing number of certain MIME types as the ranking number goes higher, except CSS. Mid-rank websites load more CSS objects. However, another observation is that the lower bound of each type reaches 0, which means not all websites need one type of content. One reason is that websites temporary outage when we visit them.

Figure 4 shows the similar result by category. Images remain the most popular objects loaded by websites. Some websites load more contents than the others for certain types, e.g. sports. The beyond average results show the demand for a variety of content in sports websites.

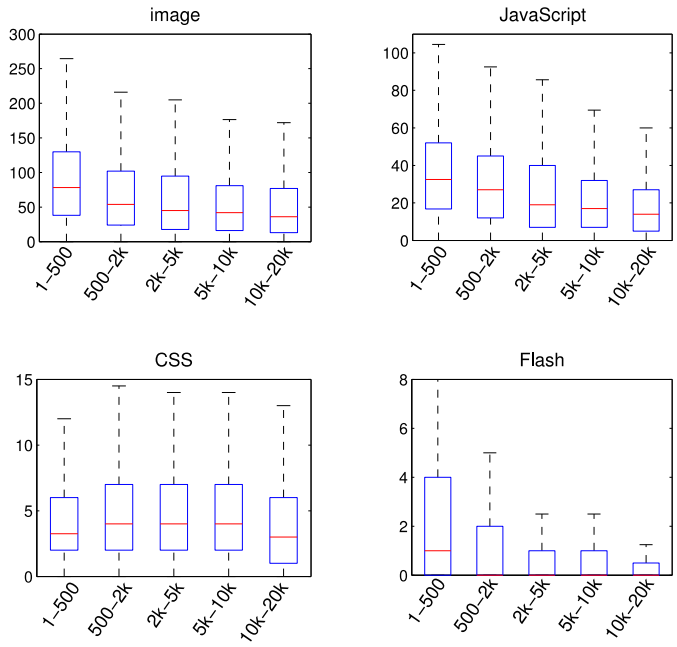


**Figure 3. Number of objects requested according to MIME types by rank**

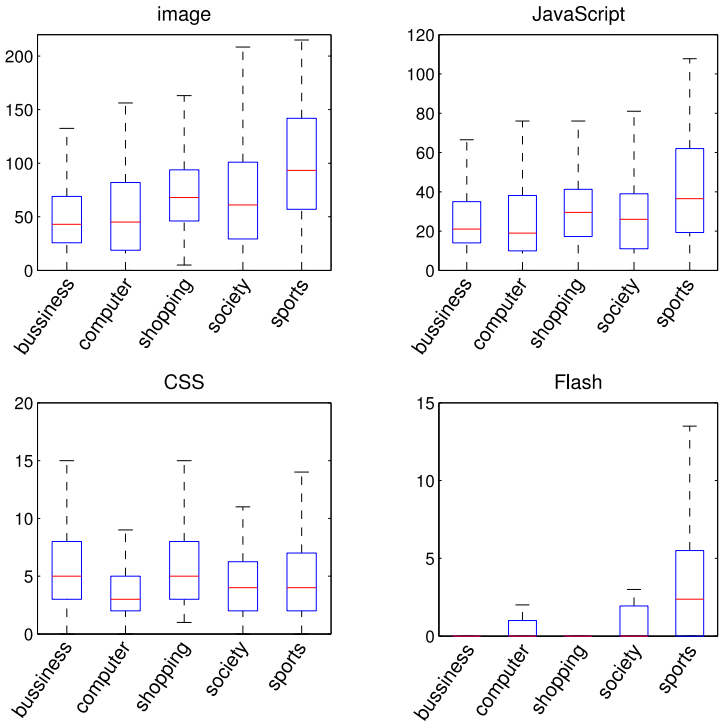


**Figure 4. Number of objects requested according to MIME types by category**

**Size of objects**: after a breakdown of requests, we take a look at the other factors of objects: **size**, because different types of objects may diverse in terms of content size. Figure 5 shows the total size of different MIME objects rendered by websites according to rank. An interesting finding is that despite more images loaded in higher ranked websites, the total size of images is more or less the same across all websites. We believe the reason is that higher rank websites load more small images, while the lower ranked websites load large images with smaller number. Flash objects are merely loaded in top rank websites with considerable size. Figure 6 shows the size of object requested by different categories. The loaded size of each object varies according to different kinds of websites. Notably, Flash values an important role in sports websites, compared with other categories.

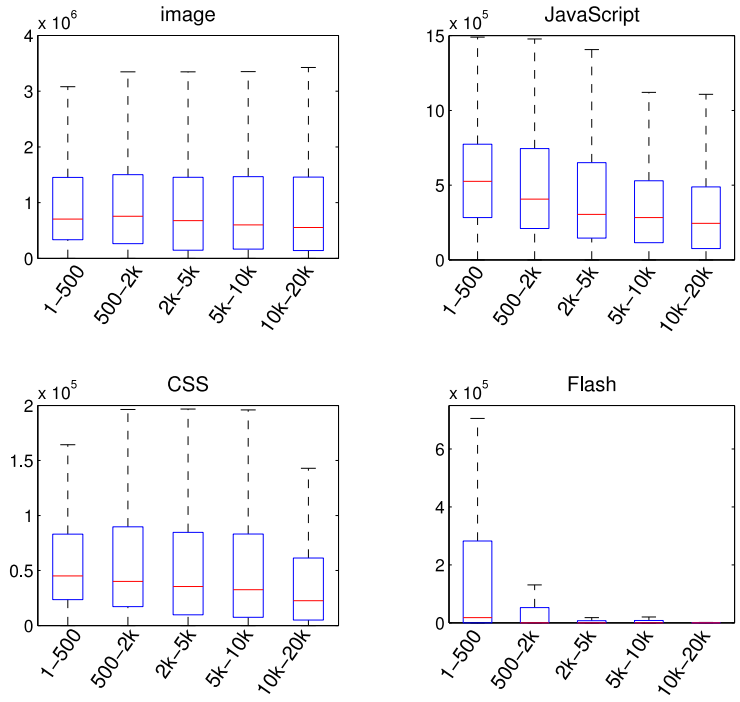


**Figure 5. Size of loaded objects according to MIME types by rank**

These results are quite similar compared to [1] with some detail differences. For example, top 500 websites load more Flash objects than other ranges, while [1] shows 401 - 1k range loads the most. We believe such difference is caused by different range selections and the way to pick websites.

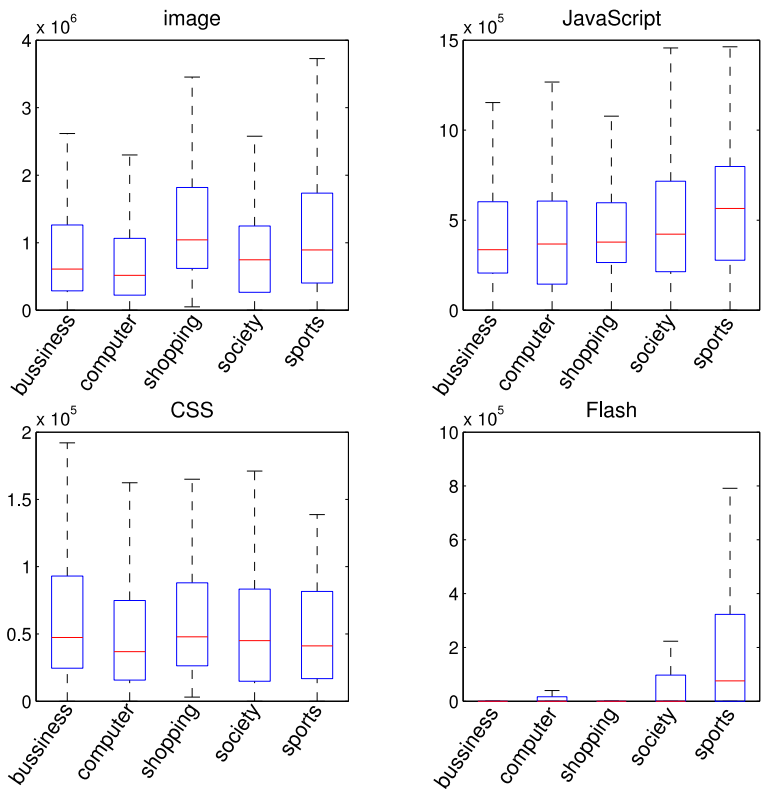


**Figure 6. Size of loaded objects according to MIME types by category**

**Overview**: Combing all types of objects together, we also want to look at the composition of the objects requested. Figure 7 shows the pie chart for all the content loaded across all websites regardless of ranking or category. Undoubtedly, images occupy the most areas of both pie charts. JavaScript and CSS take up almost the same area in both pies as well. However, Flash, which has a very small fraction, becomes the third in size excluding other types. The html-xml and text obviously shrink from number to size. The reason is straightforward. Flash is usually much larger than plain text objects such as text or html-xml.

If we look at the absolute value, Figure 8 gives the details of each content requested by websites. We can find both similarity and diversity between different MIME type objects. We are confident that more than 20% of websites sends more than 100 requests for images, the number of which exceeds any other types of requests. Followed by JavaScript, images are the most popular and important part of a web page content. Comparatively, we can also find less number of CSS and html-xml objects in websites. According to Figure 9, the size of images is still larger than the others due to large

numbers. 40% websites asks more than 1 MBytes images. Then JavaScript, Flash and CSS objects also play important roles in website loading procedure. Although a number of text objects are loaded, size of text is comparatively smaller than other types.

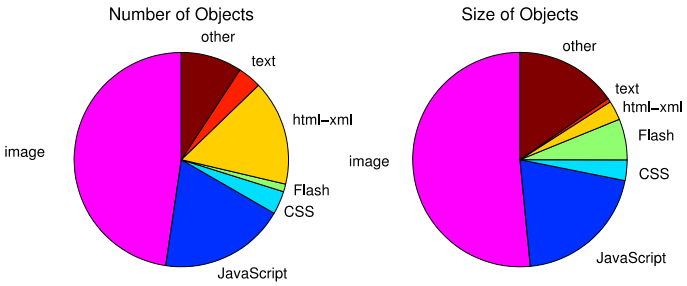


**Figure 7. Composition of MIME types according to all objects**

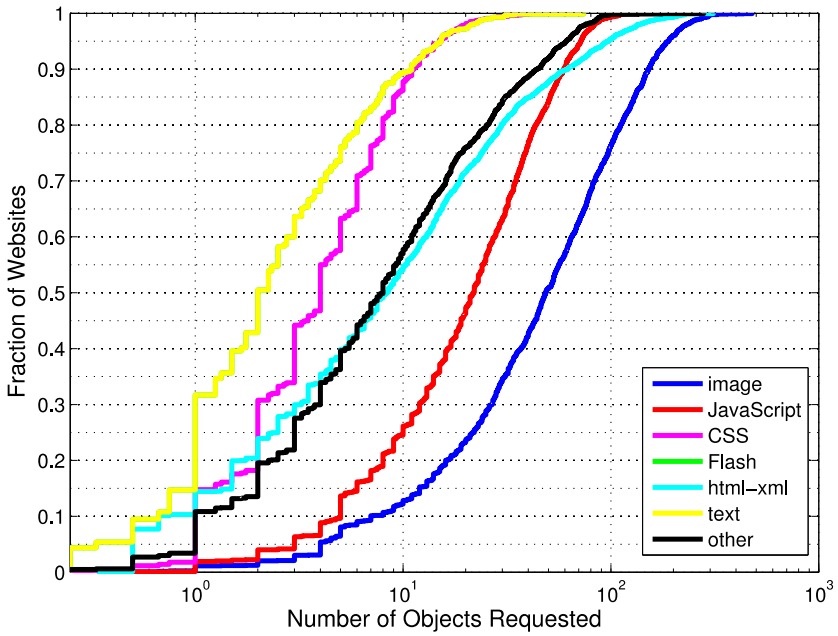


**Figure 8. Cumulative Distribution Function (CDF) plot of total number of objects according to MIME type**

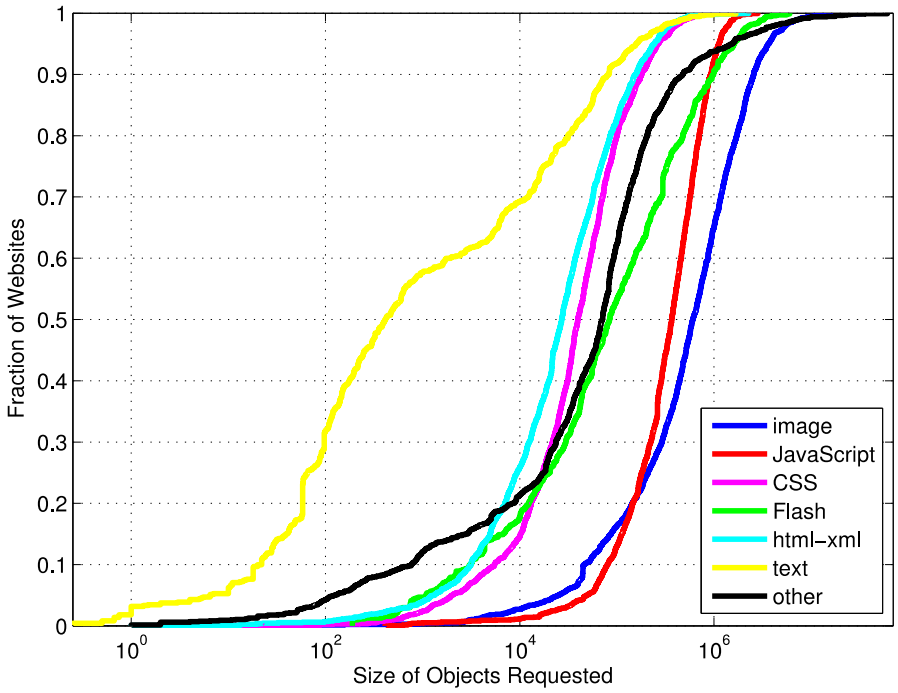


**Figure 9. Cumulative Distribution Function (CDF) plot of total size of objects according to MIME type**

### Non-origin Servers

For most websites, not all contents are provided by their own. Pictures, flash videos and others may be rendered by other servers with totally different domain names. A good example is Google ads. Lots of websites render Google ads for revenue, which are redirected to Google servers to get objects. Such method reduces the burden of original server by distributing data flows to different places. Websites may own several servers, which contribute contents and services for the same web page at same time. Investigating such distribution is interesting because we can check the trade-off relation between complexity and performance.

First of all, we mark different servers by their full domain names [1]. Each object is rendered by one particular server. Figure 10 shows the distribution of number of servers each website contacts grouped by rank. 90% of top 500 websites contact less than 120 servers. The websites with lower ranks contact much less servers. 80% of them contact less than 60-70 servers, which is almost 50% of the figure for the top 500. Figure 11 on

the other hand, illustrates the different distribution based on categories. Arts and sports websites obviously contact more servers than the other types of websites.

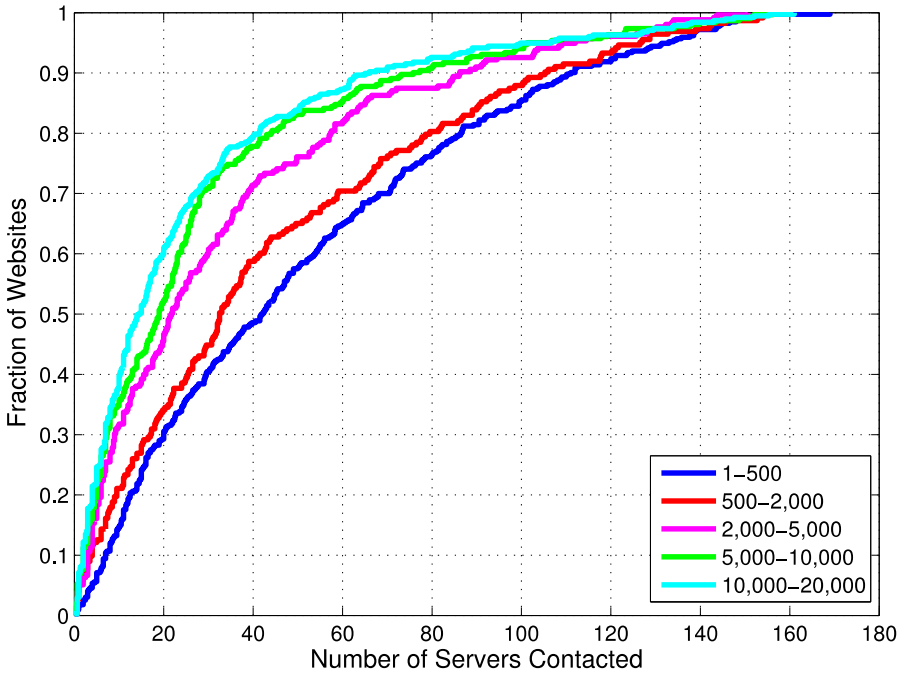


**Figure 10. Number of servers contacted by ranking**

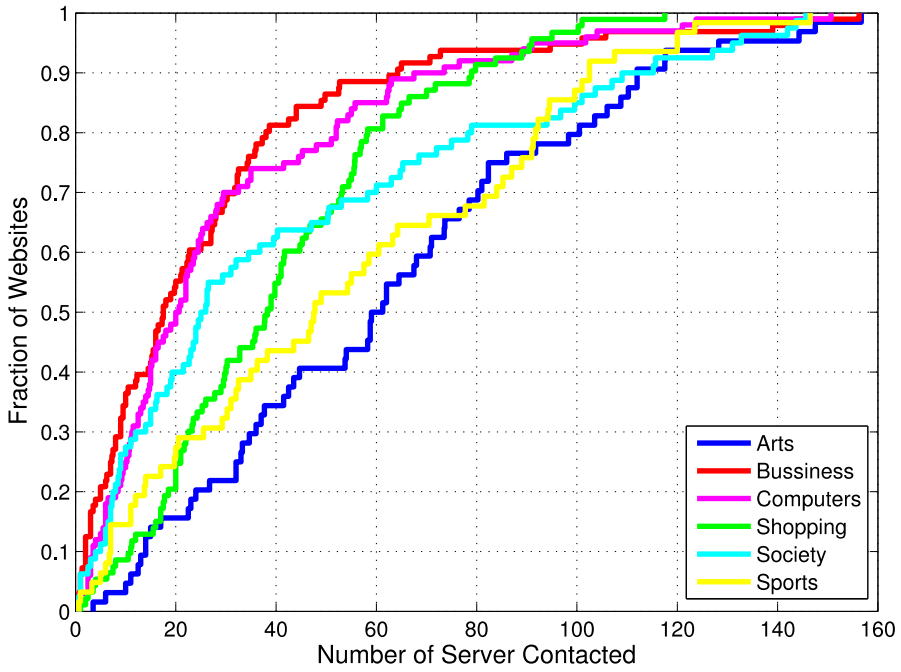


**Figure 11. Number of servers contacted by category**

However, compared to [1], the number of contacted servers significant expands. This leads the question why the number changes so much. We believe this is caused by highly mixed web connection like web APIs. For example, people may easily find many icons on *yahoo.com*, which means "sharing" to other websites, e.g. *twitter.com*. Then, we should think about where those servers belong. These servers are either origin websites or from the third party.

To find the answer, we first need to categorize different servers into clusters. If a website owns multiple servers, we need to check origin and non-origin servers [8]. An intuitive way is to observe the domain names. Obviously, *t1.google.com* and *t2.google.com* should belong to *google.com*. Some servers have different domain names, for example, *qq.com*, which is the famous website in China, owns multiple domain names including *qq.com*, *qpic.cn* and *gtimg.cn*. Visiting home page of *qq.com* sends requests to the servers with these domain names for contents and services. [8] suggests a reasonable method. A server with different domain names must share the same authoritative name server as origin server. Using such method, we can easily find that *qpic.cn* and *gtimg.cn* use *ns1.qq.com* as their authoritative name server, which is the same as *qq.com*. Hence, *qpic.cn* and *gtimg.cn* are also origin servers. Therefore, we define origin and non-origin servers by their authoritative name servers when comparing to probing URLs.

- **Origin server**: if a server shares the same authoritative name server with probing domain name server, as [8] suggests, it is considered as origin server of websites.
- **Non-origin server**: any server, whose authoritative name server is different from probing domain name server, is considered as third party of contents provider. According to [8], we mark it as non-origin server.

In our parsing process, we simply use a *dig* operation to locate its original authoritative name server. For instance, "*dig yahoo.com*" returns several lines of results, and we select servers from origin section as its answer to authoritative name server. By investigating different clusters of servers, we proceed to consider about the combination of MIME types and servers.

**Origin** servers share the same authoritative name server with the original domain name of probing URL. They are proved to be part of content providers together with

original domain name server. Figure 12 and Figure 13 present the CDF plot of number of origin servers by rank and category. We can find that most websites do not contact more than 10 origin servers regardless of rank or category.

**Non-origin servers** exceed origins significantly in terms of quantity. Figure 14 and Figure 15 give the CDF plot of non-origin servers grouped by ranking and category. Quite different from origins' figures, we find familiar curve among different ranking and categories regarding of both number and size. These figures are quite similar to Figure 12 and Figure 13 since origin servers are limited in number.

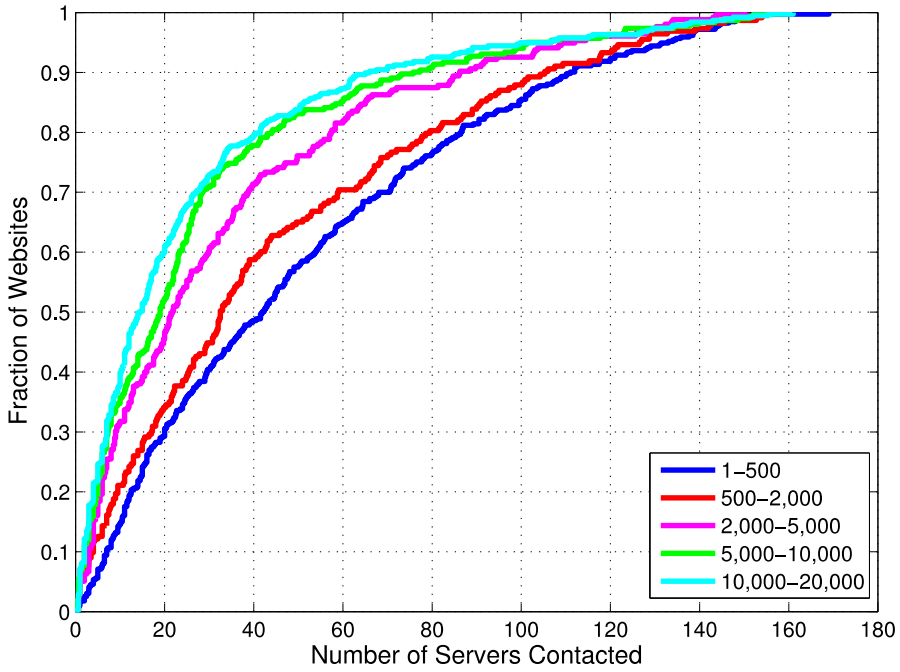


**Figure 12. Number of origin servers by rank**

If we take a look at the objects rendered by origins and non-origin servers, we can easily find their difference. Figure 16 and Figure 17 shows the composition in rendering objects by number and size. Origins mainly focus on images. They render around two third of total requests and 60% of total size, while non-origin servers are rendering more objects other than images.

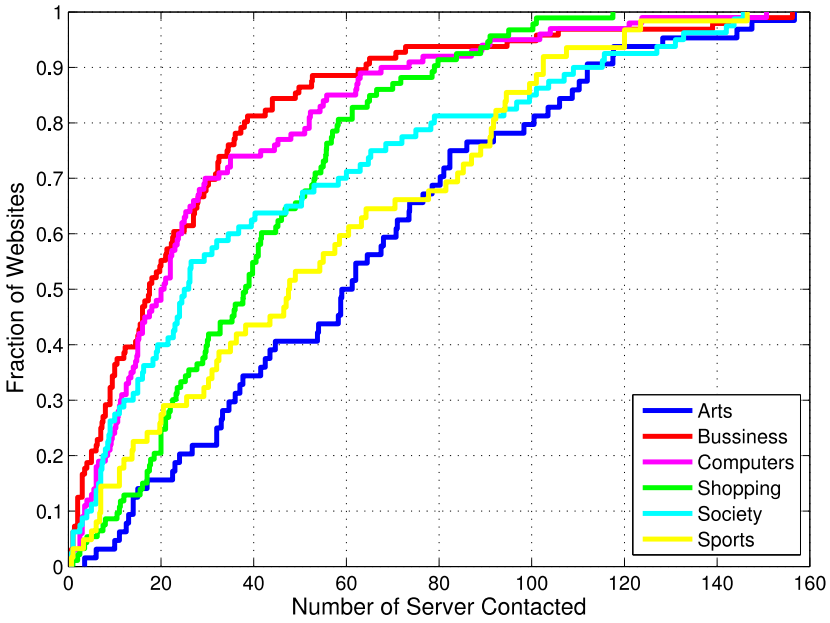


**Figure 13. Number of origin servers by category**

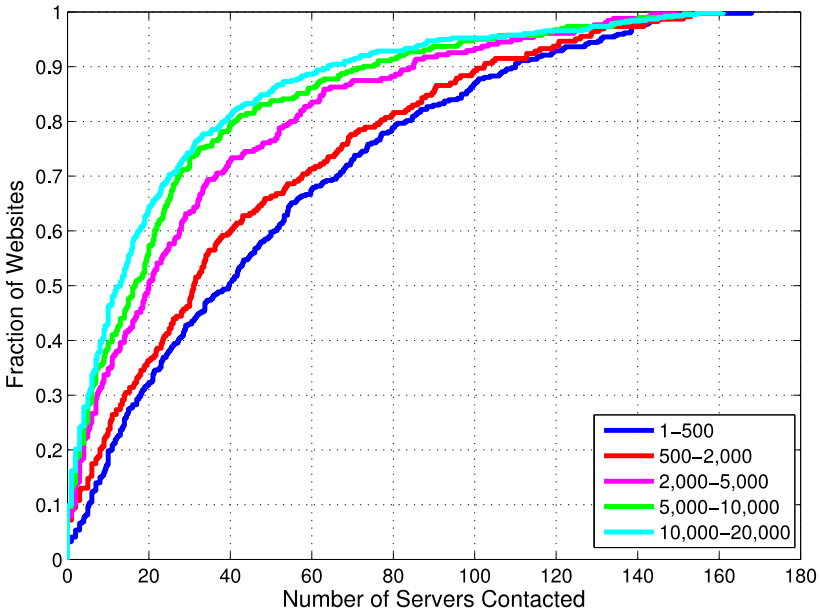


**Figure 14. Number of non-origin servers by rank**

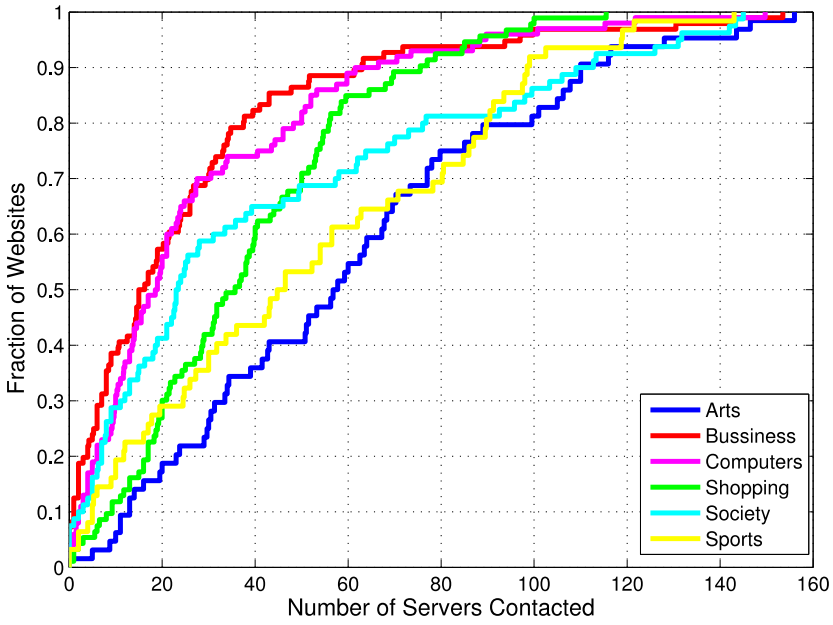


**Figure 15. Number of non-origin servers by category**

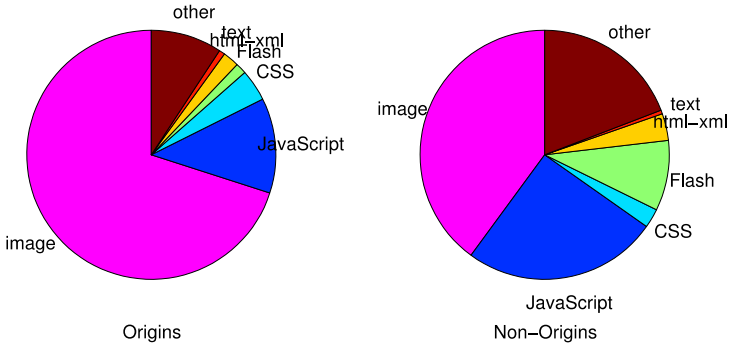


**Figure 16. Comparison of types of objects in number rendered by origin and non-origin servers**

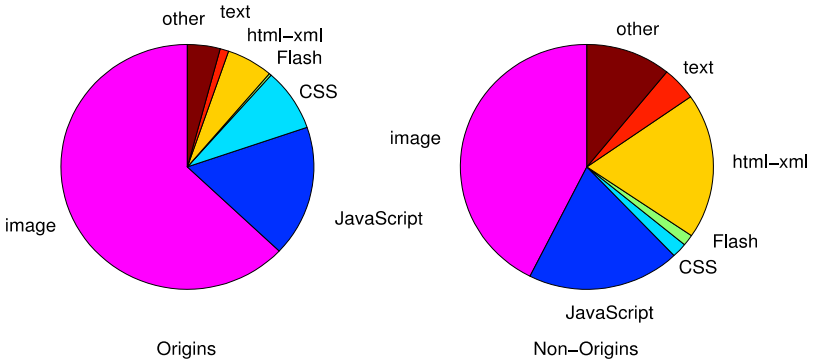


**Figure 17. Comparison of types of objects in size rendered by origin and non-origin servers**

**Revision of Methodology:**

Compared to [1], we found major difference in servers. Non-origin servers show a typical normal distribution, which is times larger than origin servers in number. In other words, websites usually contact more non-origin servers than origins. Meanwhile, the number of non-origin servers greatly increases. Another issue is that with empirical experiments, the way to identify non-origin server [8] is not reliable. Some authoritative name servers may provide huge amount of domain names, such *ns.google.com* series, though the website itself is owned by other entities.

# CHAPTER V

# PERFORMANCE

In addition to complexity, websites are also required to consider their quality of service, in which loading time takes highest priority in terms of user experience. Time consumed in page loading process is critical. We find that using **RenderEnd** as the loading time presented in HAR files in comparison is valid as [1].

## Page Loading

Figure 18 shows the mean of page loading time across all websites in milliseconds. Note that some websites are not reachable from our vantage points, and some records have invalid loading time, e.g. negative values. There are plenty of reasons. Some consume so much time so that they exceed the timeout threshold; some may be banned according to governments' policy; some values are incorrectly recorded or parsed. We found that 80% of websites load page within 10 seconds, and 50% take less than 5 seconds. Compared to [1], the loading time is more than doubled. 80% of websites takes 4-6 seconds to load in 2012. Since more contents are loaded in recent years, we think such difference is reasonable. Another factor is that the Internet traffic is also heavier. Congestion and delay can even further extend the loading time.

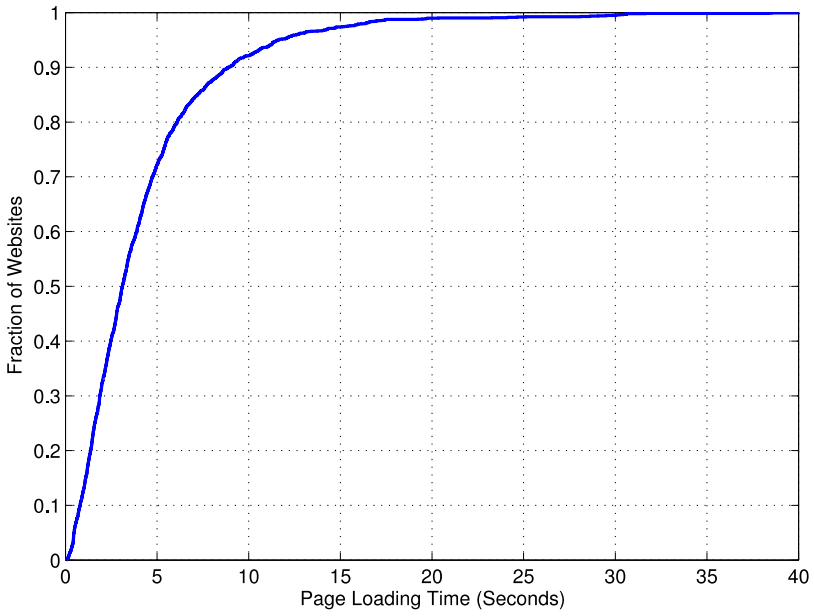


**Figure 18. Total page loading time of all vantage points across all websites**

Figure 19 shows the break down of page loading time by different categories. The results show no surprise that arts and sports websites take more time to load than other group of websites, since they contact more servers and request more objects.

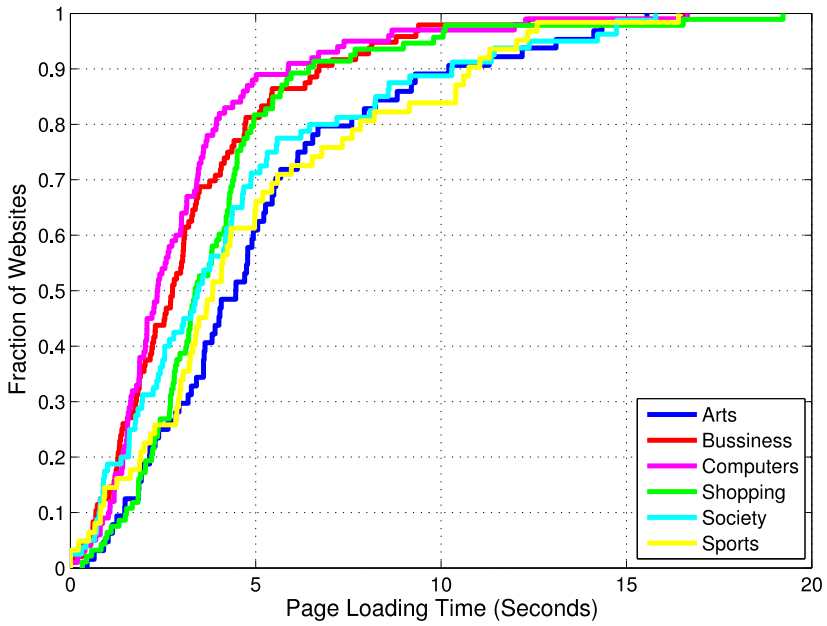


**Figure 19. Total page loading time of all vantage points across all websites by category**

### MIME Loading

Similar to complexity analysis, we consider loading time of objects for different MIME types. Figure 20 shows the loading time of websites for different MIME types. According to complexity analysis, we find the loading time for image objects is reasonable. However, comparing to number, size of the content seems less considerable when loading an object. We think this is reasonable since the overhead when establishing the communication between target server and client takes more time than loading the object itself. In other words, the more number of objects require more time before actual loading procedure.

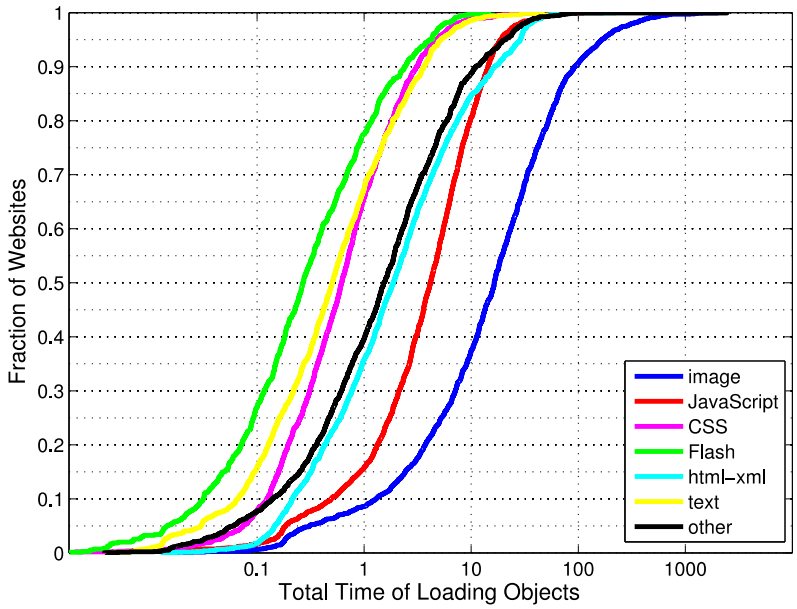


**Figure 20. Loading time of objects according to MIME types**

On the other hand, after years, estimating page time based on variety of metrics may also change. Figure 21 shows the Spearman's Correlation Coefficients by several metrics in our data set. We measure the coefficients by different vantage points we have. Since they have no particular ordering, we list some factors according to the results. Number of requests, number of servers and number of images become top 3 metrics that affect page loading time. We also find great diversity in some metrics, e.g. Size of JavaScript from Origins. They appear to be the top major factor in one particular vantage point, while no obvious correlation is identified in other vantage points.

Compared to [1], some metrics remain important, such as number of requests, number of images, number of servers and etc. Notice that we do not select the total bytes of page as one of the metrics. This metric is still important, but the page frame itself may embed numbers of objects, which are counted towards the total bytes. However, the actual value is much weaker than [1]. They also choose first and third quartile to update the correlation, which is similar to our results. Another difference is that some metrics become more or less related to page loading time, e.g. the number of other types of objects and number of JavaScript. We believe such change is caused by the updates of hardware and software. Computers become more powerful processing programs, which result in comparatively less loading time of complicated objects, e.g. JavaScript or Flash.

Meanwhile, more MIME types are obsoleted so that browsers do not recognize them any more. This causes other types more relevant to page loading time.

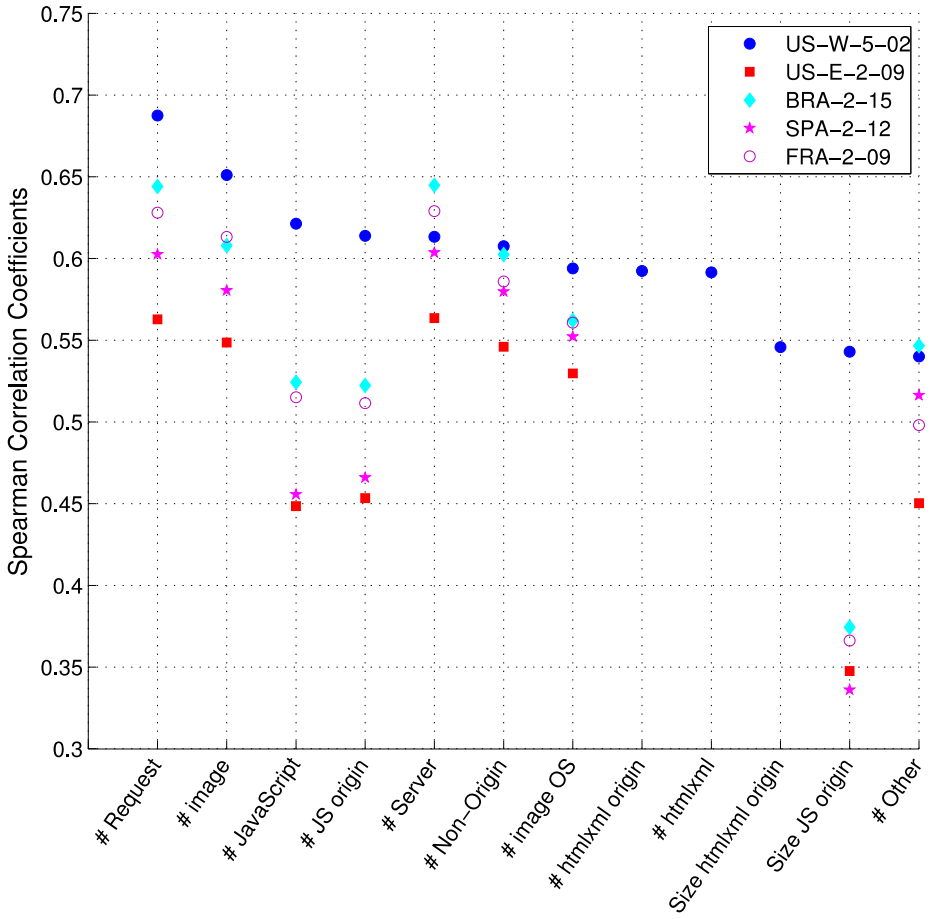


**Figure 21. Spearman Correlation Coefficients between page loading time and variety of complexity metrics**

To further confirm the correlation between the number of requests and page loading time, Figure 22 is the box-and-whiskers plot showing the correlation. The positive correlation can be concluded as well as [1] that page loading time increases with the number of objects.

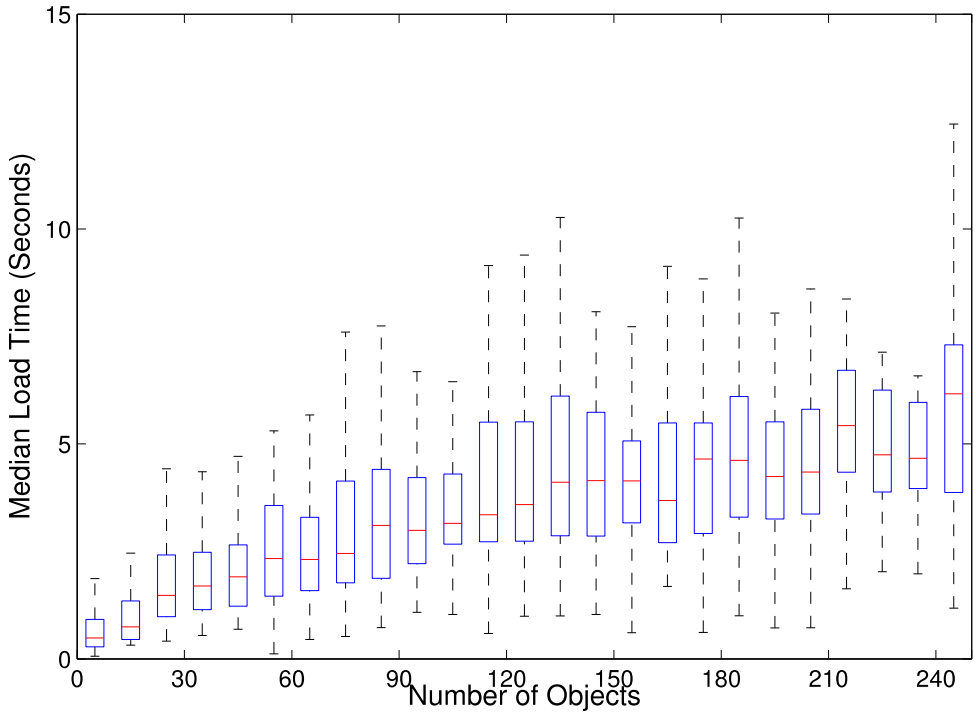


**Figure 22. Box-and-whiskers plot shows the correlation between page loading time and number of objects**

# CHAPTER VI

# COMPARISON AND CONCLUSION

The results from [1] are intuitive to direct our research. We redo the process and take a comparison after 4 years later. We conduct further analyze in each analyzing sections. The main results generally are similar with larger scale. There are more objects, more servers and more time. However, each website contacts fewer origin servers. Contents from origins are weighted less in nowadays' websites. On the other hand, page loading time is also stretched, but the conclusion from [1] stays almost the same: number of objects requested and number of servers contacted can be used to predict total page loading time.

## REFERENCES CITED

[1] Michael Butkiewicz, Harsha Madhyastha, and Vyas Sekar, "Understanding website complexity: Measurements, metrics, and implications," in IMC'11, 2011.

[2] "Selenium webdriver," http://www.seleniumhq.org/projects/webdriver/.

[3] "Firefox," https://www.mozilla.org/en-US/firefox/new/.

[4] "Firebug," http://getfirebug.com/.

[5] "Firebug extensions," https://getfirebug.com/wiki/index.php/Firebug_Extensions.

[6] "Har 1.2 spec," http://www.softwareishard.com/blog/har-12-spec/.

[7] "Harlib," https://sites.google.com/site/frogthinkerorg/projects/harlib.

[8] B. Krishnamurthy and C. Willis, "Privacy diffusion on the web: A longitudinal perspective," in WWW, 2009.